令和2年度

卒業研究報告書

付録

研究題目

Node.jsによるRGBマトリクスLED制御システム開発

システム・セットアップガイド

電気情報工学科

No.3 氏名 石川 瑛祐

指導教員

香川 福有

新居浜工業高等専門学校

目次

1 はじめに - 1 -

2 注意事項 - 1 -

3 必要な機器 - 2 -

4 システム構築 - 3 -

4.1 Raspberry Pi Imagerのインストール - 3 -

4.2 Raspberry Pi OSの書き込み - 5 -

4.3 Raspberry Pi OSのセットアップ - 7 -

4.4 ネットワークの設定 - 10 -

4.5 SSHの設定 - 12 -

4.6 led-matrixの環境構築 - 13 -

4.7 Node.jsスクリプトの永続化 - 15 -

5 配線 - 16 -

6 使用方法 - 16 -

# はじめに

本書は令和2年度に香川研究室の卒業研究『Node.jsによるRGBマトリクスLED制御システム開発』で開発されたマトリクスLEDパネルをWebブラウザからの操作で制御するシステムをセットアップするガイドブックである。

本書の手順に従えば、システムの動作環境を再現することができる。

# 注意事項

* CUIの操作について  
   本書内にはCUIで操作するコマンドやその出力結果を次のように四角の枠で囲って記述している。

$ echo test

test

先頭に$の付いている行はコマンドの行である。$は入力せずそれ以降をターミナルにそのまま入力する。

それ以外の行は標準出力への表示を表す。

コマンドは途中まで入力して残りの文字列が一意に定まることが確定しているときはTabキーを押すと自動補完が有効になる。

* viの操作

本書ではテキストファイルの編集にviを使用することを想定している。

viはCUIのテキストエディタで、次のようにして起動する。

$ vi <編集したいテキストファイル名>

上記のコマンドでviを起動するとコマンドモードでviの画面が出現する。viには主にコマンドモードと編集モードの2モードがあり、これを使い分けながらテキストを編集する。

コマンドモードの状態でiを押すと編集モードに切り替えられ、文字が入力できるように成る。編集モードからコマンドモードに戻るときはEscまたはCtrl-cを押す。以後、Ctrlキーは「C」と省略して記述する。大文字のキーはShiftを押しながら押すことを意味する。

表1に代表的なviのコマンドを示す。

表 1. viのコマンド

|  |  |
| --- | --- |
| キー | 機能 |
| :q! | セーブせずに終了 |
| ZZ | セーブして終了 |
| i | 現在のカーソル位置から入力開始 |
| r | 現在のカーソル位置の文字を置換 |
| A | 現在の行の末尾から入力開始 |
| O | 現在の前の行に入力開始 |
| o | 現在の次の行に入力開始 |
| h | 左へ移動 |
| j | 上へ移動 |
| k | 下へ移動 |
| l | 右へ移動 |
| 0 | 行頭へ移動 |
| $ | 行末へ移動 |
| yy | 1行コピー |
| p | ペースト |
| u | 直前の動作の取り消し |
| Esc(またはC-c) | 編集モードを終了 |

この他に:<数字>で指定した行番号へ移動する、/<検索対象文字列>で検索も利用できる。

操作に迷ったらまずはEscを押して:q!で編集前の状態に戻ってviを終了してやり直すことを推奨する。

# 必要な機器

以下の表2の機器を用意する。

表 2. 使用機器

|  |  |
| --- | --- |
| 名称 | 個数 |
| Raspberry Pi Model B Rev 1.2 | 1 |
| RGBフルカラードットマトリクスLEDパネル | 3 |
| フラットケーブル | 3 |
| LEDパネル用電源ケーブル | 1 |
| microSDカード | 1 |
| RGB Matrix HAT + RTC for Raspberry Pi | 1 |
| ACアダプタ | 1 |
| USB AC アダプタ | 1 |
| USBマウス | 1 |
| USBキーボード | 1 |
| HDMIケーブル | 1 |
| ディスプレイ | 1 |
| LANケーブル | 1 |

セットアップにはインターネット環境が必要である。

その他、Webブラウザが動作するPCやスマートフォンがあればWebシステムを遠隔操作することができる。

以下、特筆すべき機器について注意事項を示す。

* Raspberry Pi Model B Rev 1.2

Raspberry Pi 3 Model B以降のモデルに対応する。

* RGBフルカラードットマトリクスLEDパネル

HUB75規格対応のパネルが必要。16x32ドットに限る。

* フラットケーブル

LEDパネルの制御用ケーブル。HUB75規格対応品。

* LEDパネル用電源ケーブル

4pin電源ケーブルが必要。

* microSDカード

16GB程度以上推奨。

* ACアダプタ

HATへの電源供給用。5[V]で6.5[A]以上のものが必要。

* USB ACアダプタ

Raspberry Piへの電源供給用。2.5[A]以上推奨。

# システム構築

## Raspberry Pi Imagerのインストール

Raspberry PiにOSをインストールする。OSをSDカードに書き込むためにRaspberry Pi Imagerを使用する。

https://www.raspberrypi.org/software/にアクセスして、Install Raspberry Pi OS using Raspberry Pi Imagerの章から、Download for <使用しているOS名>をクリックして、Raspberry Pi Imagerをダウンロードする。以下の図1は上記URLの画面である。

グラフィカル ユーザー インターフェイス, Web サイト

自動的に生成された説明

図 1. Raspberry Pi ImagerのWebサイト

ダウンロードしたimager\_1.5.exeを実行し、インストーラに従ってインストールする。インストール完了後、起動すると以下の図2のような画面が出現する。

グラフィカル ユーザー インターフェイス, Web サイト

自動的に生成された説明

図 2. Raspberry Pi Imager

## Raspberry Pi OSの書き込み

PCにmicroSDカードを挿入する。

Raspberry Pi ImagerのOperating System以下のCHOOSE OSをクリックすると図3のような画面が現れるので、OSタイプを選択する。今回はRaspberry Pi OS (32-bit)を選択する。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図 3. Raspberry Pi Imager – CHOOSE OS

次にSD Card以下のCHOOSE SD CARDをクリックして出現する図4の画面からインストール先のSDカードを選択する。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図 4. Raspberry Pi Imager – CHOOSE SD CARD

ここまで選択が完了したら、WRITEボタンを押して図5のようなSDカード内のデータが削除される警告にYESを押して承諾してOSを書き込む。

グラフィカル ユーザー インターフェイス, アプリケーション, Web サイト

自動的に生成された説明

図 5. Raspberry Pi Imager - WRITE

SDカードへの書き込みが終わったら、図6のようなメッセージが出るのでCONTINUEを押してSDカードを取り外す。

グラフィカル ユーザー インターフェイス, アプリケーション, Web サイト

自動的に生成された説明

図 6. Raspberry Pi Imager – Write Successful

## Raspberry Pi OSのセットアップ

Raspberry Piに4.2で作成したSDカードを装着する。

Raspberry Piにディスプレイとマウス、キーボード、LANケーブルを接続して、microUSBから電源を取って起動する。このとき、まだHATは接続しない。電源投入後最初は以下の図7のような画面が現れる。

夕日に映る建物

自動的に生成された説明

図 7. Raspberry Pi OS初回起動画面

Nextを押して設定を進めていく。

図8のSet CountryではCountryはJapan、LanguageはJapanese、TimezoneはTokyoを選択し、必要に応じてUse English LanguageやUse US Keyboardにチェックを入れる。

テキスト が含まれている画像

自動的に生成された説明

図 8. Set Country

次の図9のChange Passwordではデフォルトユーザのpiにパスワードを設定する。デフォルトではraspberryが設定されているが、脆弱なので任意のパスワードを設定する。設定できたらNextを押す。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図 9. Change Password

図10のSet Up Screenでは必要に応じてチェックを入れてNextを押す。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

図 10. Set Up Screen

Wi-Fiに接続する場合は図11のSelect WiFi Networkで接続するSSIDを選択する。今回はSkipを選択する。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ, メール

自動的に生成された説明

図 11. Select WiFi Network

図12のUpdate Softwareではシステムのアップデートを行うのでNextを選択する。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図 12. Update Software

## ネットワークの設定

Setup Completeが表示されたらGUIでの設定は完了である。Doneを押してウィンドウを閉じる。

ここからはCUIで設定を進めていく。画面左上部のターミナルのアイコン(図13)をクリックしてターミナルを開く。



図 13. ターミナルのアイコン

IPアドレスを固定する。接続するネットワークの構成に応じてここに記載する手順ではない方法でも臨機応変に対応されたい。

まず、現在のIPアドレスを確認するために$ ifconfigを実行する。実行すると以下のような結果が表示される。

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 169.254.152.200 netmask 255.255.0.0 broadcast 169.254.255.255

inet6 fe80::db23:2cc:5936:8cdf prefixlen 64 scopeid 0x20<link>

ether b8:27:eb:05:2c:3a txqueuelen 1000 (Ethernet)

RX packets 12380 bytes 613138 (598.7 KiB)

RX errors 0 dropped 2 overruns 0 frame 0

TX packets 910 bytes 67132 (65.5 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 544 bytes 51718 (50.5 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 544 bytes 51718 (50.5 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether b8:27:eb:50:79:6f txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

この結果のeth0部分のinet、及びnetmaskの部分を記憶して次の手順に進む。IPアドレスの固定を行うために/etc/dhcpcd.confを編集する。viで編集する場合は次のコマンドで行う。

$ sudo vi /etc/dhcpcd.conf

ここからはviの操作に則って解説していく。

ファイルを開いたらGを押してファイルの末尾に移動する。Oを押してファイル末尾に次の内容を記述する。

interface eth0

static ip\_address=172.19.210.25/16

static routers=172.19.1.254

static domain\_name\_servers=8.8.8.8

記述できたらC-cを押して入力モードを解除する。ZZを押して保存してviを終了する。

$ sudo reboot nowでRaspberry Piを再起動する。

再起動後はネットワークの疎通確認とIPアドレスの固定の確認を行う。

疎通確認の方法は$ ping google.comの実行やWebブラウザから任意のWebサイトが閲覧できるかなどで行う。

IPアドレスが固定できているかの確認は$ ifconfigを実行してeth0のinetの値が設定した値になっていることで行う。以下のようにifconfigの結果が変わっていたらIPアドレスの固定化に成功している。

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.19.210.25 netmask 255.255.0.0 broadcast 172.19.255.255

inet6 fe80::db23:2cc:5936:8cdf prefixlen 64 scopeid 0x20<link>

ether b8:27:eb:05:2c:3a txqueuelen 1000 (イーサネット)

RX packets 5016 bytes 4671356 (4.4 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 1699 bytes 172578 (168.5 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

## SSHの設定

SSH(Secure SHell)はネットワークを介してコンピュータを遠隔操作するための手段であり、以後Raspberry Pi上にSSHサーバを建ててセットアップをクライアントから行えるようにする。

* Raspberry Pi側の操作

画面左上のRaspberry Piアイコンをクリックし、メニューを展開した先の設定>Raspberry Piの設定を選択する。Raspberry Piの設定の中のインターフェイスタブを選択し、SSHを有効にする。SSHを有効に設定した画面が以下の図14である。

グラフィカル ユーザー インターフェイス

自動的に生成された説明

図 14. Raspberry Piの設定 > インターフェイス

有効にしたらOKを押してウィンドウを閉じる。

* クライアント側の操作

<WindowsでのSSH接続方法を書く。>

## led-matrixの環境構築

本研究で開発したシステムはGitHubリポジトリの名前を取って”led-matrix”としている。ここからはled-matrixを実行可能な環境を構築していく。作業はSSH接続したRaspberry PiのターミナルまたはRaspberry Piへの直接の操作によって行う。

まず、パッケージマネージャaptをアップデートする。以下のコマンドを実行する。

$ sudo apt update

$ sudo apt upgrade

アップデートが終わったらNode.jsをインストールする。以下のコマンドを実行する。

$ sudo apt install -y nodejs npm

インストールが終わったらNode.jsのバージョンを確認する。

$ node -v

v10.23.1

令和3年2月18日時点では標準でインストールされるNode.jsは10.23.1であった。

v10系でも動作するとされているが、開発時に使用したv12.18.3の使用を推奨するためNode.jsのバージョン管理ソフト『n』を導入していく。以下のコマンドを実行する。

$ sudo npm cache clean

$ sudo npm install npm n -g

$ sudo n 12.18.3

以上のコマンドが実行できたら$ vi ~/.bashrcでnでインストールしたNode.jsへパスを通す。記述する内容は以下の通りでこれを末尾に追記する。

export N\_PREFIX=$HOME/.n

export PATH=$N\_PREFIX/bin:$PATH

以上の作業を終えてNode.jsのバージョンが変わっているか確認する。

$ node -v

v12.18.3

以上の表示が出たらNode.jsの導入は完了である。

次にGitHubからled-matrixリポジトリをダウンロードする。

まず、$ cdを実行してホームディレクトリに戻る。次に以下のコマンドでホームディレクトリ直下にled-matrixをダウンロードする。

$ git clone https://github.com/xiexxa/led-matrix.git

mariaDBをインストールする。mariaDBはシステムの履歴の管理などに使用しているデータベースであり、動作に必要不可欠である。以下のコマンドでインストールする。

$ sudo apt install mariadb-server

インストールが完了したら、mariaDBのセットアップを行う。以下のコマンドでmariaDBのセットアップウィザードを起動する。

$ sudo /usr/bin/mysql\_secure\_installation

様々な質問が聞かれるが、全てEnterまたはyで回答する。途中rootのパスワードを設定する場面ではパスワードを覚えておく。

初期設定が完了するとled-matrixから使用するmariaDBのユーザを作成する。以下のコマンドでmariaDBにrootとしてログインする。

$ sudo mariadb

ログインできたら以下のSQLを実行する。

create user eisuke identified by ‘password’

eisukeというユーザをパスワードはpasswordとして作成している。これがled-matrixからデータベースに接続するためのIDとなる。

同時にこのユーザに権限を付与する。本来であれば細かく権限を分けたほうが良いが今回はすべての権限を与える。以下のSQLを実行する。

create database testdb;

grant all on \*.\* to 'eisuke'@'%';

testdbというデータベースを作成し、eisukeユーザにすべての権限を与えている。

exitを実行してmariaDBからログアウトする。

次に、led-matrixのシステムに必要なテーブルを作成する。$ cd ~/led-matrix/sqlを実行してSQLファイルが保存されているディレクトリに移動する。

ここで、$ mariadb testdb -u eisuke -p < init.sqlを実行してSQLファイルをデータベースに反映させる。

実行が終わったらSQLファイルが適用されているか確認する。以下のコマンドでeisukeユーザとしてデータベースにログインする。

$ mariadb -u eisuke -p

ログインできたら以下のSQLを実行する。

use testdb;

show tables;

testdb内のすべてのテーブルが表示される。以下のように表示されていればデータベースの設定は完了である。

+------------------+

| Tables\_in\_testdb |

+------------------+

| feeds |

| phrases |

| text\_histories |

+------------------+

led-matrixの依存モジュールをインストールする。本システムは複数のNode.jsモジュールの組み合わせで構成されるため依存性の解決が必要である。

まず、$ cd ~/led-matrixでled-matrixのディレクトリに戻る。そして以下のコマンドを実行し依存しているモジュールをまとめてインストールする。

$ sudo npm install

このコマンドは少し完了まで時間がかかるので、ここで5章の配線を進めていても構わない。

ただし、easybotics-rpi-rgb-led-matrixは上記のコマンドではインストールされないことが多いようである。そのためこのモジュールのみ別でインストールする必要があるので以下のコマンドを実行する。

$ npm i easybotics-rpi-rgb-led-matrix

インストールが完了したら$ node server.jsを実行して3000と表示されることを確認する。エラーが出なければ必要なモジュールはインストール出来ている。確認できたらC-cでスクリプトを終了する。

フロントエンドのソフトウェアをビルドする。$ npm run buildを実行するとled-matrixディレクトリ内のdist/にプロダクションバージョンのファイルが出力される。

ビルドが終わったら$ node server.jsを実行してWebブラウザでhttp://<Raspberry PiのIPアドレス>:3000にアクセスすると図15のようにled-matrixのトップ画面が閲覧できる。

コンピューターのスクリーンショット

自動的に生成された説明

図 15. led-matrix – トップ画面

## Node.jsスクリプトの永続化

server.jsがRaspberry Piの起動時に自動的に起動するように設定する。

Node.jsのスクリプト永続化にforeverをインストールする。

$ sudo npm install -g forever

起動時にforeverでled-matrixのスクリプトが実行されるように登録するため、/etc/rc.localを編集する。以下のコマンドでviを起動する。

$ sudo vi /etc/rc.local

以下の内容をファイル末尾のexitの上に記述する。

forever start /home/pi/led-matrix/server.js

記述できたらZZで保存して終了する。

次回起動時以降、led-matrixが自動起動する。

# 配線

ハードウェアの配線の方法について解説する。

最終的に、以下の図16のように配線する。

回路 が含まれている画像

自動的に生成された説明

図 16. 配線完成図

はじめから手順を追って解説していく。

1. LEDパネルをフラットケーブルで接続する

LEDパネルの裏面左右にあるINとOUTの端子同士をフラットケーブルで図17のように接続する。

電子機器の部品

低い精度で自動的に生成された説明

図 17. フラットケーブルの接続

1. LEDパネル用電源ケーブルを接続する

LEDパネルに4pinコネクタの電源ケーブルを接続する。ツメがコネクタに付いているのでLEDパネル裏面中央部の4pin端子に図18のように接続する。

メーター が含まれている画像

自動的に生成された説明

図 18. 電源ケーブルの接続

1. HATにフラットケーブルと電源を接続する

まずHATに電源ケーブルを接続していく。HATのOutput 5Vの部分に②で接続した電源ケーブルの一対を+は赤、-は黒のケーブルに対応するように図19のように接続する。

回路 が含まれている画像

自動的に生成された説明

図 19. HATへの電源ケーブル接続

次にフラットケーブルをHATに接続する。HAT中央のHUB75の端子にフラットケーブルを接続し、一対は①で接続したLEDパネルのIN端子に図20のように接続する。

パソコンの画面

中程度の精度で自動的に生成された説明

図 20. HATへのフラットケーブル接続

1. HATをRaspberry Piに接続する

Raspberry Piに重ねるように、HATを載せてRaspberry PiのGPIOピンに図21のように接続する。

電子機器の部品

中程度の精度で自動的に生成された説明

図 21. HATのRaspberry Piへの接続

1. ACアダプタを接続する

Raspberry PiのmicroUSB端子にUSB ACあ打プタの電源を、HATの電源端子にACアダプタの電源を差し込み、通電する。

以上で配線は完了である。

# 使用方法